"Method and circuit for noise estimation, related filter, terminal and communication network using same, and computer program product therefor"

* * *

5   Field of the invention

The present invention relates to techniques for noise estimation.

Specifically, the invention relates to techniques for determining, in a noise reduction process applied

10 to a signal, for example a speech signal, affected by background noise, an update function relating a new value of estimated noise power (P_{noise_New}) with a previous value of estimated noise power (P_{noise}).

The invention was developed by paying specific

15 attention to the possible application to noise estimation in short time spectral amplitude methods, such as subtraction-type methods (also known as spectral subtraction methods).

Spectral subtraction is a method for enhancing the

20 perceived quality of speech signals in the presence of additive noise such as ambient or background noise. Spectral subtraction encompasses a variety of related and derived methods.

The Wiener filter is exemplary of a filter

25 implementing this type of methods and adapted for use with the invention, wherein the update function is a function of the previous estimated noise power (P_{noise}) and a mean input power spectral density (P_{in_PSD}).

In the following, reference will thus be primarily

30 made to a Wiener filter. Those of skill in the art will however appreciate that the invention is not limited to Wiener filters but applies in general to all those types of techniques that require noise estimation along the same lines of noise estimation in Wiener filters.

35

2

Description of the related art

Nowadays, users of cell phones require an ever-increasing number of service utilities and the number
5 of functions available in cell phones is correspondingly increased. Voice recognition plays a key role as a means to improve the human being/machine interface and make the communication process easier.

In this context, ETSI (European Telecommunication
10 Standards Institute) released a standard for a Distributed Speech Recognition System containing an Advanced Front-End Feature Extraction Algorithm intended to cope with background noise in a more robust manner (ETSI ES 202.050). A basic element
15 distinguishing this standard over its predecessors is the Noise Reduction (NR) block: one of the main features of this block is the use of a Wiener filter.

The Wiener filter is the subject of extensive literature, as witnessed e.g. by the following patent
20 documents that discuss the use of a Wiener filter for noise reduction in acoustic signals:

    EP-A-1 278 185,

    EP-A-1 132 896,

    US-B-6 445 801,
25   US-B-6 415 253,

    U.S. patent applications US 2003 033 139, and 2003 0018 471 A1.

Specifically, the noise reduction algorithm proposed in the ETSI standard is a combination of a
30 two-stage Wiener filter with another processing technique whose features are of no momentum for the purpose of the instant application.

Based on the related theory, an input noisy signal passes through the two stages of the Wiener filter,
35 that are similar but not identical, to produce a de-

noised output signal. The transfer function of the filter in the frequency domain weighs the spectrum as a function of the signal-to-noise ratio (SNR) of the input signal. For that purpose, the algorithm uses
5   noise estimation: this is developed as a function of time, the mean (or average) input signal power spectral density and the previously estimated noise power.

In the second filter stage, noise estimation is performed differently from the first stage, by using a
10   complex function to calculate an "update" variable that should be multiplied by a previously estimated noise power figure to compute a new power figure.

Embedded systems, as used e.g. in mobile phone terminals, usually incorporate limited memory and
15   processing resources. Real-time applications such as noise-reduction therefore impose strict time constraints on such embedded systems. For that reason a distributed approach is considered in the ETSI standard referred to in the foregoing.

20   In addition, for those applications, fixed-point notation is currently used in the place of floating-point notation since, e.g. i) the hardware of such embedded systems is mostly inadequate to support floating-point operation and ii) fixed-point notation
25   is much faster to run, despite the loss in accuracy and some additional controls required.

For noise estimation of the second stage in a Wiener filter, three multiplications and two divisions are needed for each power sample and their equivalents
30   in fixed-point notation must be implemented. These operations, and especially division, are very cumbersome to implement. Real-time systems therefore attempt to avoid such operations.

4

Object and summary of the invention

While quite a few well-known methods exist for
computing a division result in fixed-point arithmetic,
5   the need is still felt for arrangements that may permit
fast and efficient noise estimation (e.g. according to
the   standard   mentioned   previously).   This   while
retaining the possibility of operating with fixed-point
arithmetic   and   striking   a   sensible   balance   between
10   accuracy of the results and speed of computation, by
acting on few parameters.

The object of the present invention is to satisfy
the needs considered in the foregoing.

According to the present invention, that object is
15   achieved by means of a method having the features set
forth in the claims that follow. The invention also
relates to a corresponding circuit and encompasses a
noise reduction filter and a communication terminal
including   such   a   circuit,   a   communication   network
20   comprising   such   a   terminal,   and   a   computer   program
product loadable in the memory of at least one computer
and   comprising   software   code   portions   for   performing
the steps of the method of the invention when the
product is run on a computer. As used herein, reference
25   to such a computer program product is intended to be
equivalent to reference to a computer-readable medium
containing   instructions   for   controlling   a   computer
system to coordinate the performance of the method of
the invention. Reference to "at least one" computer is
30   obviously intended to highlight the possibility for the
arrangement of the invention to be implemented in a de-
centralized fashion.

A preferred embodiment of the invention is thus a
circuit   for   determining,   in   a   filter   for   noise
35   reduction   in   a   signal,   such   as   a   speech   signal,

affected by background noise, an update function relating a new value of estimated noise power with a previous value of estimated noise power. The update function is a function of the previous estimated noise
5   power and a mean input power spectral density, and the circuit includes a look-up table having stored therein values for the update function as well as an input module for a current value for the mean input power spectral density. Search circuitry is associated with
10 the look-up table for selectively searching values for the update function in the look-up table using the previous value of estimated noise power and the current value for the mean input power spectral density as a first and a second entry for the search.
15   Preferably, the search circuitry is configured for performing the search in the look-up table on the basis of an index computed starting from the first and second search entries.
        Such an arrangement takes advantage of the fact
20 that the contour lines of the expression used for noise estimation e.g. in the second stage of a Wiener filter are essentially straight lines with the origin at the zero and variable angular coefficients.
        In such a preferred embodiment, the value of the
25 "update" variable can thus be calculated as a function of the angle formed in a corresponding surface by the average input power spectral density and the noise power estimated previously, the value for the "update" variable being essentially constant for that given
30 angle.
        The arrangement described herein thus dispenses with complex operations such as division by substituting therefor simple and fast addition, subtraction and shift operations. This without any
35 appreciable loss in terms of accuracy of the results.

6

The arrangement described herein also strikes a reasonable balance in terms of accuracy of the results obtained and computational effort, thus achieving a great efficiency in different specific implementations.

5   A preferred application of the arrangement described herein is speech processing in a Wiener filter as defined by the ETSI Standard ES 202.050 (oftentimes referred to as "Aurora" standard).

A particularly preferred application is noise 10 reduction for speech processing in mobile/embedded terminals. These applications require low cost/real time equipment, and benefit from the fine-tuning of accuracy of the results and the speed of computation offered by the arrangement described herein.

15   The invention therefore fulfils the need for arrangements that permit fast and efficient noise estimation, while retaining the possibility of operating with fixed-point arithmetic and a good flexibility in balancing accuracy of the results and 20 speed of computation, by acting on few parameters.

Brief description of the annexed drawings

The invention will now be described, by way of example only, with reference to the annexed figures of drawing, wherein:

25   - figure 1 shows the contour lines of a function as determined computed in the arrangement shown herein,

- figure 2, comprising four diagrams designated 2a to 2d, respectively, shows various alternative arrangements for determining the function shown in 30 figure 1,

- figure 3 is a block diagram of the arrangement described herein,

- figure 4 shows the possible application of the arrangement of figure 3 within the framework of a 35 mobile terminal in a communication network, and

- figure 5, 6 and 7 are block diagrams detailing operation of the arrangement shown herein.

Detailed description of preferred embodiments of the invention

An exemplary embodiment of the arrangement described herein will now be detailed by referring to a preferred application to a computationally critical part of a Wiener filter used in noise reduction for speech processing.

Specifically, reference will be made – by way of example – to the procedures defined in the standard designated ETSI ES 202.050 v 1.1.1 (2002-10) Speech processing, transmission and Quality aspects (STQ); Distributed Speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms

The arrangement described herein performs noise estimation, by means of the so-called "update" function in the Wiener filter. This is defined by equations 5.10 of ETSI Standard ES 202.050 and represents the most complex part of the Wiener filter. This is essentially related to the need of performing complex operations, such as divisions and multiplications, which are rather heavy to perform in terms of computational load.

By way of theoretical introduction, the "update" function of the Wiener filter is computed in the second stage of noise reduction and is described in analytical form by the following equation:

$$update = 0.9 + 0.1 \cdot \frac{P_{in\_PSD}}{P_{in\_PSD} + P_{noise}} \cdot \left(1 + \frac{1}{1 + 0.1 \cdot \frac{P_{in\_PSD}}{P_{noise}}} \right) \qquad (1)$$

where P_{in_PSD} is the mean (average) input power spectral density, P_{noise} is the previously estimated noise power.

8

The entity designated "update" is used in periodically computing a new value of estimated noise power, P_{noise_New}, (related to the samples transmitted in the current time interval), in the second stage of the
5   Wiener filter.

Specifically, the following relationship applies:

$$P_{noise\_New} = update \cdot P_{noise} \quad (2)$$

10   In principle, well known methods exist for replacing complex arithmetic operations (e.g. division) with fixed-point algorithms that require simpler operations such as shifts, additions and subtraction. Direct verification shows that these methods are not
15   efficient for certain specific computations as those required by the "update" function represented by the relationship (1) above.

Even without wishing to be bound to any specific theory in that respect, Applicants have reason to
20   believe that this inefficiency is intrinsically related to the nature of the processes involved, namely the stochastic processes represented by the signal, in particular a speech signal, and the ambient (or background) noise signal. Specifically, the variables
25   P_{in_PSD} and P_noise can assume a wide range of values, and both their values and their ratio cannot be represented with high precision in fixed-point arithmetic and requires an expensive dedicated floating-point unit. Additionally, the known methods referred to in the
30   foregoing require the use of algorithms that, in addition to being cumbersome, do not permit control of the accuracy of the results and of the complexity of computation.

The arrangement described herein is based on the
35   unexpected recognition of the following.

While militating against the effectiveness of well known methods for replacing complex arithmetic operations with simpler fixed-point operations, the very nature of the speech signal and the ambient (or

5   background) noise and of the "update" function makes it possible to replace the entire expression for the "update" function with simple processing steps having associated a minimum effort of computation while preserving a high accuracy in the final results.

10   In fact, the applicability of the arrangement described herein is in no way limited to the definition of the update function provided by formula (1) reported in the foregoing.

     Those of skill in the art will thus promptly
15   appreciate that, while demonstrated in the following with specific reference to the relationship (1) above, the arrangement described herein is applicable to any "update" function - even if different from the relationship (1) above - that is adapted to calculate,

20   (for instance in a spectral subtraction filter or a Wiener filter), a new value of estimated noise power, P_{noise_New}, referred to the samples transmitted in the current time interval (these samples having a mean power spectral density P_{in_PSD}) , starting from a

25   previously estimated noise power, i.e. P_{noise}, when such function is in the generic form:

$$update = f\left(\frac{P_{in\_PSD}}{P_{noise}}\right) \quad (1.a)$$

30   where f() is a generic function of the ratio P_{in_PSD}/P_{noise}, without any limitation about its complexity. The ratio P_{in_PSD}/P_{noise} is also known as "Signal to Noise Ratio" or SNR, and equation (1.a) can also be written in the equivalent form:

$$update = f(SNR) \quad (1.b)$$

For instance, the previous equation (1) belongs to
5   the set of functions (1.a) because it can be rewritten
as:

$$update = 0.9 + 0.1 \cdot \frac{\dfrac{P_{in\_PSD}}{P_{noise}}}{\dfrac{P_{bi\_PSD}}{P_{noise}} + 1} \cdot \left(1 + \frac{1}{1 + 0.1 \cdot \dfrac{P_{in\_PSD}}{P_{noise}}}\right) \quad (1.c)$$

or in an equivalent form as:

10

$$update = 0.9 + 0.1 \cdot \frac{SNR}{SNR + 1} \cdot \left(1 + \frac{1}{1 + 0.1 \cdot SNR}\right) \quad (1.d)$$

By considering (as a non limiting example) the
relationship (1), equivalent to (1.c) and (1.d) above,
15   one can demonstrate that, as a two-variable function,
dependent on P_{in_PSD} and P_{noise}, the function in question,
or any analogous function that can be expressed in the
form (1.a), describes a surface that have contour lines
represented by straight lines (with different angular
20   coefficients and non-uniform spacing) having origin at
the intersection of the axes. The update values
corresponding to these contour lines are not subject to
specific restrictions. The contour lines of this
surface are shown in figure 1.
25   The arrangement described herein provides for
partitioning this surface in constant-value regions
separated by straight lines with well-known angular
coefficients. Of course, the only significant quadrant
is the one that have both P_{in_PSD} and P_{noise} positive,
30   because these values are the powers of physical signals
and cannot assume negative values.

Instead of being calculated analytically (with the associated computational load), the values for "update" can thus be simply searched in a predefined table such as a look-up table (LUT) addressed by means of an index

5   value that unambiguously identifies one of the angular regions defined in the foregoing. The index value in question is defined on the basis P_{noise} and P_{in_PSD} that represent a first and a second entry for the search. Each region in the surface is related to a pre-

10 calculated "update" value that minimizes the maximum error.

An increased number of regions will result in higher accuracy, while also requiring a larger look-up table and more iterations to find the correct region.

15   In embedded software/hardware design, a trade-off needs to be reached between memory requirements, speed of computation and precision, and an optimal configuration reaches such a trade-off.

Deeper analysis of figure 1 shows that in that

20 portion of the surface where P_{in_PSD} is high and P_{noise} is low, the contour lines are very close to each other. This means that in that portion there is a more evident and steep variation of the "update" function in comparison with the other portions of the surface.

25   A two-step approach may thus be devised comprised of a general approximation for the whole surface and a more refined search for the more critical, steeper area. That area is usually a very populated area (many significant points in it), as the signal is much higher

30 than the noise, and therefore deserves a special attention.

The first approximation step, used over the whole surface, takes a parameter N and divides the surface into $2 \cdot N$ angular regions, which are the areas between

35 the straight lines:

12

$$y = j \cdot x \, / \, N \qquad \text{when} \quad P_{in\_PSD} < \quad P_{noise}$$
$$y = N \cdot x \, / \, j \qquad \text{when} \quad P_{in\_PSD} >= P_{noise} \qquad (3)$$

5   with j assuming the integer values from 1 to N, with P_{noise} plotted on the X-axis and P_{in_PSD} on the y-axis (as shown in the figures 1 and 2).

This is essentially a linear approximation, since the intervals between the lines on the y-axis for the
10   first group of lines, or on the x-axis for the second group, are constant and their value is always x/N or y/N, respectively.

Another approximation is essentially an exponential approximation and can be used as an alternative to the
15   one described previously in order to obtain a good approximation of the "update" function, especially in the critical area. The line with the higher angular coefficient (slope) traced by the linear approximation in the whole surface is the line y = N·x and the
20   approximation in the critical area subdivides the region included between that line and the y-axis in a plurality of sub-regions. For this critical area another parameter, NH, is used, independent from N.

Within the critical area, any of the two
25   approximations, linear or exponential, can be used. The choice depends on the specific formulation of the update function that in its generic formulation is expressed in the form (1.a).

Using the linear approximation again here, there
30   will be more NH sub-regions in this critical area, which are determined by the lines with equation:

$$y = NH \cdot N \cdot x \, / \, j \qquad (4)$$

35   with j from 1 to NH-1, integer values.

13

Using the exponential method, there will more NH+1 sub-regions inside that region, determined by the lines with equation:

5       $y = 2^j \cdot N \cdot x$ (5)

with j assuming the integer values from 1 to NH.

The diagrams of figures 2a to 2d reproduce the funtion of figure 1 by further highlighting the postion
10 of the line y = N·x.

As indicated, such line partitions the surface representative of the function into two regions or areas where two different approximation methods may be used.

15 Specifically, figure 2a refers to an embodiment where a linear approximation is used for the area on the right hand side of the line y = N·x. Conversely, in the remaining – critical – portion on the left hand side of the line y = N·x the exponential approximation
20 expressed by formula (5) above is used. This is better appreciated by referring to figure 2b, which is a reproduction of the critical area in question wherein the abscissa scale has been expanded.

Figure 2c shows an embodiment where a linear
25 approximation is again used for the area on the right hand side of the line y = N·x. In this case, for the remaining – critical – portion on the left hand side of the line y = N·x a linear approximation as expressed by formula (4) above is used. This is again better
30 appreciated by referring to figure 2d, which is a reproduction of the critical area in question wherein the abscissa scale has been expanded.

Formulas (4) and (5) are not by themselves mutually exclusive: in fact they can be used
35 concurrently, by further partitioning the "critical"

14

area considered in the foregoing into sub-areas. For each sub-area either of the two formulas can be used, by defining respective independent values for NH, namely NH1 ed NH2.

5   In connection with the specific formulation (1) (or its equivalent (1.c), handling the critical zone as a single zone is generally satisfactory. In the case of the formulation (1.a), applying different approximation methods may be advantageous. The area   of major
10   interest being in any case the area where the steepest changes in the update function are located.

A practical implementation of the arrangement described herein is explained in detail in the following.

15   A circuit architecture adapted to compute the "update" function of the Wiener filter (according to the ETSI "Aurora" Standard) is shown in figure 3.

The architecture of figure 3, indicated as a whole as 50, is suitable to be included (figure 4)- in a
20   manner known per se e.g. as an embedded system associated - in a noise reduction block 40 that also contains a noise subtraction filter or a Wiener filter 55 and in turn associated to speech processing apparatus 60.

25   Such apparatus - known per se - may be included in the Speech recognition Front-End of a system (e.g. according to the ETSI ES 202.050 standard), associated with a mobile terminal TM in a mobile communication network. All the information signals processed therein
30   can be represented using a fixed-point notation.

In the following, a set of speech signal samples (as transmitted e.g. in a 10 ms time interval) will be identified as a "frame".

Corresponding   processing   steps   will   now   be

described by referring primarily to the block diagram
of figure 3 while also referring to the flow charts of
figures 5 to 7.

Specifically, the following description represents
5 the entire flow of figure 5, including a binary search
(block 120 of figures 5 and 7) and the functions for
determining an index as used for addressing a look-up
table (blocks 124 and 128 in figure 7). In practice,
all the functions on the right hand side of figure 7
10 coincide with the blocks 210 e 212 reproduced in the.
left hand side of figure 7. For the sake of simplicity,
figure 3 groups in a single block some functions shown
separately in figures 5 e 7.

The input information that is processed consists
15 of:

- P_{in_PSD}, or Mean Power Spectral Density, produced -
in a manner known per se - using the data sample in the
current frame and in the previously transmitted frame
(block 200 in figure 5);

20 - P_{noise}, or Noise Spectrum Estimate, referred to the
samples transmitted in the previous time interval
(block 209 in figure 5).

The two power values (unsigned values) are compared
in a Value Switch Compare block 10, and the result of
25 the comparison is used (in a step 100) to select one of
two sections of the "update" function projection to
apply the interpolation. The two sections correspond
respectively to the graph areas with $P_{in\_PSD} < P_{noise}$ (see
102 in figure 5) and with $P_{in\_PSD} > P_{noise}$ (104 in figure
30 5); each of these sections is subsequently divided into
angular regions (see equations 3).

The larger of the Mean Power Spectral Density and
the Noise Spectrum Estimate is then used to compute a

16

first coordinate ("Coord") in a Compute First Coord block 12. This is essentially done in steps 110a to 110d in figure 5 (steps 106 and 108 will be detailed in the following).

5   The simplest way to perform this operation is a 1-position right shift in a fixed-point arithmetic (corresponding to a division by 2 steps 110a, 110b); the value obtained becomes the first coordinate and is used to compute (steps 112a to 112d of figure 5) the

10   first value of the "Increment" variable, the latter operation being performed in a Compute New Increment block 14.

       The two quantities "Coord" and "Increment" are used to find a region defined by two contour lines that

15   approximate the value of the "update" function. As shown previously, the contour lines of the "update" function are straight lines having their origin at the intersection of the axis (P_{in_PSD}, P_{noise}); such property is used to find an approximation of the "update"

20   function, defining angular regions and using a fast search performed on one of the two input quantities, P_{in_PSD} and P_{noise}.

       As indicated, in order to compute the best approximation of the "update" function, initially two

25   sections are identified: $P_{in\_PSD} > P_{noise}$ and $P_{in\_PSD} < P_{noise}$ (102 and 104, respectively, in figure 5); after the definition of the section, the angular regions of equations (3) are considered in performing the search for the best approximation for the "update" value.

30   A binary search (or an equivalent search procedure) is then applied to modify the "Coord" value by a quantity equals to the "Increment" variable. Such operations are iterated a number of times defined by an Iteration Counter block: the simplest implementation of

this block is a counter that will count the number of cycles needed by the search algorithm to find the region that leads to the best approximation of the "update" function value. Each value of the Iteration 5 Counter 16 will thus correspond to a cycle of the search algorithm. The Iteration Counter 16 is also used to select the first value of the "Coord" and "Increment" values (i.e when Iteration Counter = 0) for any new frame.

10 The lower value of the Mean Power Spectral Density and the Noise Spectrum Estimate becomes a "Target" value (11), blocks 118a and 118b in figure 5 and the purpose of the search is to find the angular region that contains this value. At each cycle, a comparison 15 is carried out between the "Coord" value (line 13 in figure 3) and the "Target" value (line 11 in figure 3) in a compare block 18; this comparison allows iterating a convergence of the "Coord" value toward the Target, updating accordingly the "Coord" value.

20 The compare block 18 is used to determine if the value of "Coord" (stored in a Store Coord block 20) is strictly less than, greater than or equal to the "Target" value (line 11 in figure 3).

The convergence is achieved in a Compute New Coord 25 block 22 in figure 3.

The values of "Coord" (as stored in a block 20) aand "Increment" (as stored in a Store Increment block 24) as well as the output of the Compare block 18 are used into the Compute New Coord (block 22) to compute 30 the new value of "Coord" for the next cycle.

The value of the Iteration Counter (block 16) is also considered in order to stop the operations when the maximum number of iterations (steps 116a-116d in figure 5) has been reached.

In its simplest form, the Compute New Coord block 22 will add or subtract "Increment" to "Coord" according to the output of the Compare block 18. The value of "Increment" is decreased (usually divided by two, using a 1-bit right shift) at each cycle, through the Compute New Increment block 14.

This procedure is used for the linear approximation, obtained applying equations (3) and (4).

As indicated, a different way of computing "Coord" for the next cycle is the exponential approximation.

In that case the block 22 sets the new value of "Coord" (line 13 of figure 3) equal to "Increment", instead of adding or subtracting "Increment" from the "Coord" value. In that case the search function can stop before reaching the maximum number of iterations, depending on the result of comparison between "Coord" and "Target", typically as soon as "Coord" is less than "Target". Such procedure is used for the exponential regions definition, obtained applying equation (6).

At each iteration, the output of the Compare block 18, is sent to the Compute Index block 26. This block (26) also receives the output of the Iteration Counter block 16, used to start and stop the computation for any new Frame, and the output 17 of Value Switch Compare block 10 (that is the result of the comparison $P_{in\_PSD} > P_{noise}$), used at the Iteration zero.

This information is used to recursively compute an "Index" value; inputting them to a Compute Index block 26 thus contributes to build a portion (tipically a bit) of "Index" value at each iteration. This "Index" value (line 19 in figure 3) unambiguously identifies the angular region that gives the best approximation of the "update" function.

In its simplest form, the Index is a binary word of L bits, the number of regions used in the interpolation process being equal or less to 2^L. Each bit of this word will then correspond to a result of the Compare

5   block 18 or of the Value Switch Compare block 10, available on line 17 in figure 3 (that is the result of the comparison $P_{in\_PSD} > P_{noise}$) during each of the cycles executed by the search procedures.

The partial determined value for Index at each

10   iteration is stored in a Store Index block 28.

When the search procedures are completed, according to the Iteration Counter 16 (for the linear approximation) or the output of the Compare block 18 (for the exponential approximation), the final value of

15   "Index" is used to access a table (represented by a LUT memory block 30 in figure 3).

The LUT Memory 30 returns the approximated value of the "update" function for the region unambiguously identified by the "Index" value.

20   By resorting to the architecture shown in figure 3 the procedure for computing the "update" functions can be completed using only shift, addition, subtraction and comparisons steps of binary values expressed in fixed-point notation.

25   Complex operations are avoided and computation is speeded up, while the accuracy of the final result can be easily controlled. The error in the approximation is controlled by the number of regions used in the search procedures; a trade-off between speed and accuracy (or

30   complexity of the logic implementation) can be achieved acting on the number of regions which directly defines the total number of cycles required to compute the final "Index" value, thus to retrieve the approximated value of the "Update" function.

In order to achieve the best results with a minimum computation effort, various search functions can be implemented to refine the result at the desired accuracy.

5   One solution that leads to a high accuracy without having to consider an excessive number of regions, is to find a first approximation by using few angular regions, then to refine the result applying a more accurate search inside the region found. Such an

10   approach is focused on the angular regions that contain a bigger number of contour lines of the "update" function. Experimental analysis may permit to locate nearly optimal regions sizes that effectively do not lead to loss of performance in the application.

15   For Wiener filter design, such analysis can be done at the output of the filter or preferably at the output of the Noise Reduction Filter that includes a Wiener Filter implementing the equations (1) and (2).

Experimentation has shown that the approximation of

20   the region with the highest P_{in_PSD} / P_{noise} ratio is critical in terms of the result accuracy; at the same time, such a "critical region" has the highest gradient of the "update" function. To improve the quality of the results, the approximation method is refined inside

25   this critical region.

A nearly optimal trade-off between accuracy and computation effort is found by applying the previously described approach using two distinct angular regions definitions: in a first phase, a region is found; if

30   the result is the region with the highest P_{in_PSD}/P_{noise} ratio, then a refining step is applied, searching a better approximation using more dense sub-regions. This second phase can use a different search function, for instance using exponential approximation, instead of

35   the linear one.

A fast and accurate implementation is obtained defining:

$$N = 2^K \qquad (6)$$

5

N being the number of angular regions for each of the 2 semi-areas, respectively identified by $P_{in\_PSD} > P_{noise}$ and $P_{in\_PSD} < P_{noise}$, and K any positive integer value.

10   The search procedure uses the linear approximation method previously described. Using this method, the critical region is easily identified by the relationship:

15    $$P_{in\_PSD} > P_{noise} * 2^K \qquad (7)$$

This region definition (that lead to the other result in step 106 of figure 5) is especially advantageous, because it allows using binary shifts

20   instead of multiplication, with a fixed-point arithmetic.

As a consequence, the comparison $P_{in\_PSD} > P_{noise} * 2^K$ is performed in the Value Switch Compare block 10 (in addition to the comparison $P_{in\_PSD} > P_{noise}$). According

25   to the result of these comparisons, the Compute First Coord block 12 will execute the operation $P_{in\_PSD} / 2^K$ (performed as a right shift of N positions in step 110d) or a simple 1 bit right shift (division by 2 steps 110b) to compute the initial "Coord" value with a

30   fixed-point notation.

The result of the comparison $P_{in\_PSD} > P_{noise} * 2^K$ is also forwarded to the Compute Index block 26 and Compute New Coord block 22. These can thus apply different search procedures if the "update" function

35   value belongs to the critical region.

22

The overall computing resources required for
properly managing the critical region are thus
moderate, while the accuracy can be easily controlled,
defining NH dense sub-regions inside the critical
5   region, with NH being totally independent from N.

Approximation within the critical region can be
carried out using the linear or the exponential
approximations previously described. If the linear
approximation is used also in the critical region
10   (depending on the outcome of step 108 in figure 5),
then the most efficient definition of NH is:

$$NH=2^P \qquad (8)$$

15   being P any positive integer value, independent
from K.

Due to the similarity with expression (6), the
search function applied to sub-regions in the critical
region is the same as the procedure applied to search
20   any other region.

If exponential approximation is used, NH can be any
positive integer value; in such case the architecture
of figure 3 is still valid and, within each cycle, the
only difference is in the operation executed by the
25   Compute New Coord block 22, as previously explained.

The architecture proposed in figure 3 is thus
adapted to perform different operations in the Compute
First Coord block 12 and Compute New Coord block 22,
according to the identification step of the critical
30   region (see equation 7) carried out in the Value Switch
Compare block 10, and transmitted with a signal over a
line 21 of figure 3.

Effectiveness of the arrangement described herein
can be tested by comparing the total frame noise energy
35   variable on a time scale, for a reference model,

obtained in a traditional manner (using divisions), and for the arrangement described herein, which only uses simple operations.

5 In this way, it is possible to observe that the noise energy, computed from equations (1) and (2), is influenced by this approximated solution and validate the results over the estimated noise power. Experimentation shows that the arrangement described herein can be used for optimizing accuracy or the Look-
10 up-table size and number of iterations.

A good trade-off was shown to be obtained with N = 16 and NH = 6 in the exponential mode.

Another advantage of the arrangement described herein lies in that it uses simple operations in a
15 fixed-point arithmetic. This arrangement is thus ideally suitable for low-cost devices, such as embedded systems for consumer electronics and for low/moderate processing power, real time equipment, like those used for mobile communication, where a high level of
20 accuracy can be obtained by using computation of low complexity.

Figure 6 shows a practical application of the arrangement described herein to noise estimation within a voice sample processor, such as the noise reduction
25 block included in the distributed speech recognition front end as defined by the ETSI 202.050 Standard.

After a start step 1000 and a step 1002 devoted to LUT and parameter initialization, voice samples are grouped over time intervals or slots (such group of
30 data is also defined as a "frame").

The steps of figure 6 are thus repeated regularly for all the frames, at regular time intervals. In a step 1004, a check is made as to whether new signal samples, e.g. speech signal samples,
35 are available for processing.

24

In the negative (no new samples available), a
buffer flushing step 1006 is performed.

If, conversely, new samples are available, after
a possible pre-processing step 1008 (of a known type),
5   the value of P_{in_PSD} is determined in a step 1010.

Thereafter the parameter P_{noise_New} is estimated in
a step 1012 as detailed in the following in connection
with figure 7.

Therafter, filtering (of any known type among
10 the various techniques considered at the outset of the
present description) is applied in a step 1014, and the
filtered samples are buffered in a step 1016.

Step 1018 marks the end of processing for a
given frame, and the sequence of steps described in the
15 foregoing is then repeated for a new frame.

As indicated the new value for P_{noise_New} (valid
for the present time interval), is determined based on
the mean input signal power spectral density (P_{in_PSD})
and the previous value of P_{noise}.

20   Figure 7 shows the steps used to compute the new
value of the noise power estimation P_{noise_New}, by means
of the computation of the "update" variable.

At the beginning of the Voice Stream (First Data
Set, 202 in figure 7), the noise power estimation is
25 initialized (step 204 in figure 7) and the parameters
of the proposed algorithm are set (step 206 in figure
7). For all the subsequent frames, the previous noise
power estimation is considered (step 208 in figure 7).

Such value (designated 209 in figure 7) is used,
30 together with the value P_{in_PSD} (200 in figure 7), to
find the best approximating region in the plane
containing the contour lines of the "Update" function
(210 in figure 7).

The identification of the region found is then
35 used to retrieve the value of the "Update" function

25

(212 in figure 7) in the look up table 130 of figure 3.
The value of "Update" thus retrieved is used to compute
the value of P_{noise_New} (214 in figure 7) that will be
used in the next time interval (216 in figure 7).

5   These operations, as shown in the left-hand side
of the flow chart of figure 7 are implemented through
the instructions shown by the boxes on the right side
of the same figure.

          The most significant of those instructions are
10   also shown in the flow chart of figure 5.

          More specifically, following the flow of data
processing, the approximating region is found by
executing the following sequence of instructions:

          - i) select the X-Y section of the Update
15   surface projection (100 and 104 in figure 7 and figure
5);

          - ii) check the positioning in the critical sub-
region, depending on result of i) (operations 106 in
figure 5);

20   - iii) initialize the variable Coord (operations
110x) and Increment (operations 112x) according to
parameters (linear or exponential approximation, 108)
and the results of the previous instructions;

          - iv) perform a binary search is subsequently
25   applied (120);

          - v) the region found is then associated to an
index (124);

          - vi) such index is used to address (step 128)
the table of values or LUT table 30 of figure 3.

30   Of course, without prejudice to the underlying
principle of the invention, the details and embodiments
may vary, also significantly, with respect to what has
been described, by way of example only, without
departing from the scope of the invention as defined in
35   the claims that follow.